

Development, verification, and maintenance of computational software in geodynamics

MICHAEL GURNIS, WALTER LANDRY, EH TAN, LUIS ARMENDARIZ,
LEIF STRAND, AND MICHAEL AIVAZIS

5.1 Introduction

Research on dynamical processes within the Earth and planets increasingly relies upon sophisticated, large-scale computational models. Improved understanding of fundamental physical processes such as mantle convection and the geodynamo, magma dynamics, crustal and lithospheric deformation, earthquake nucleation, and seismic wave propagation, are heavily dependent upon better numerical modeling. Surprisingly, the rate-limiting factor for progress in these areas is not just computing hardware, as was once the case. Rather, advances in software are not keeping pace with the recent improvements in hardware. Modeling tools in geophysics are usually developed and maintained by individual scientists, or by small groups. But it is difficult for any individual, or even a small group, to keep up with sweeping advances in computing hardware, parallel processing software, and numerical modeling methodology.

We will focus on the challenges faced by computational geophysics and the response of a community initiative in the United States called the Computational Infrastructure for Geodynamics (CIG). Instead of reviewing all of the activities CIG has been involved with, we will focus on just a few so as to describe the multiple ways that a virtual organization developed and used software within the rapidly evolving backdrop of computational science. We will focus on the scientific topics of mantle convection, tectonics, and computational seismology, although CIG has also been deeply involved with magma dynamics and the geodynamo.

Mantle convection is at the heart of understanding how the Earth works, but the process remains poorly understood at best. Progress on fundamental questions, such as the dynamic origin of tectonic plates, layering and stratification within the mantle, geochemical reservoirs, the thermal history of the Earth, the interpretation of tomography, and the source of volcanic hotspots, all require an interdisciplinary approach. Numerical models of mantle convection must therefore assimilate

information from a wide range of disciplines, including seismology, geochemistry, mineral and rock physics, geodesy, and tectonics.

The technical challenges associated with modeling mantle convection are substantial (Moresi *et al.*, 2000). Mantle convection is characterized by strongly variable (i.e., stress-, temperature-, and pressure-dependent) viscosities (Schubert *et al.*, 2001). In addition, the mantle is chemically heterogeneous, is replete with silicate melts and volatiles, and has numerous pressure- and temperature-induced structural (phase) changes that affect its dynamics (Davies, 1999). Equally importantly, we have learned that it is not practical to model mantle convection using a wholly theoretical or “first-principles” approach, even if we had the hardware resources at our disposal. Two decades of experience have taught us that “first-principle” mantle convection models cannot replicate the unique history of the Earth, starting from arbitrary initial conditions (Bercovici *et al.*, 2000). Accordingly, mantle convection models need to be more closely integrated into the “real world” through direct data assimilation and by direct testing against a variety of observations (Davies, 1999).

The lithosphere, with the embedded crust, represents the main thermal boundary layer of the Earth’s heat engine and, as such, encompasses a wide range of pressure and temperature conditions with a diversity of deformational mechanisms (Ranalli, 1995). On million-year timescales, the crust is a storehouse of observational constraints drawn from structural geology, tectonics, metamorphic petrology, stratigraphy, etc., that have been acquired for more than a century. Recently, deep seismic profiling, receiver-function analysis, and magnetotelluric sounding have greatly increased our understanding of crustal and lithospheric structure, and the USArray component of EarthScope is flooding us with new observations. Numerical modeling has become an essential step in the integration of these data into process-orientated models of orogenesis, lithospheric stretching, sedimentary basin genesis, and plate boundary deformation (Karner *et al.*, 2004).

A rapidly advancing area of crustal geodynamics, one of great societal importance, is the problem of the physics of the earthquake cycle. Because of the recent development of the capability for high-accuracy geodetic monitoring of crustal deformation in real time using GPS and InSAR, this field, long starved for data, is now a burgeoning observational science. Recent observations made with high-precision space geodesy indicate that displacements caused by slow aseismic motions following earthquakes can be comparable to co-seismic displacements, demonstrating substantial post-seismic evolution of strain and stress in addition to co-seismic changes. Existing GPS arrays in the Pacific Northwest, southern California, Japan, and Taiwan, and the Plate Boundary Observatory component of EarthScope, are providing heretofore-unavailable observational constraints on the stress and strain changes through the earthquake cycle.

Deformation of the lithosphere presents a number of challenges to numerical simulation. The lithospheric mantle encompasses a differential temperature of up to 1400 °C and an effective viscosity contrast of many orders of magnitude. The continental crust exhibits compositional heterogeneity and low-temperature domains dominated by frictional plastic deformation. The complex physics of frictional materials is particularly challenging because it involves strain-localization, time- and rate-dependent yield strength, and strain softening. Faults are ephemeral on a million-year timescale and thus form or vanish according to the evolution of stress and strain. Crustal deformation is a free-surface problem and sensitive to the complexities of the Earth's surface, including physical and chemical erosion, mass transport by rivers and ocean currents, and deposition of sediment. Ultimately, assimilation of data such as the state of stress in the crust or fault rupture histories from paleoseismic data will provide important constraints on the dynamics of the system. Thus, modeling fault dynamics promises new breakthroughs in earthquake hazard mitigation by uniting various earthquake observations (data assimilation) into comprehensive treatments at the regional (fault zone) level. This integration must cross timescales of seconds to weeks (fault afterslip) to years (transient and steady viscoelastic relaxation) to geologic time (development of fault systems), as well as spatial scales of hundreds of kilometers (fault systems) to centimeters (fault gouge zones).

Finally, seismology is in a position akin to crustal geodynamics: It is being buried by data. The tremendous success of programs like IRIS (Incorporated Research Institutions for Seismology) in the acquisition and archival of seismic data now provides more than 25 Tbytes of high-quality digital seismograms available instantly over the Internet. New data are now coming in at a rate of 8 Tbytes/year with the USArray component of EarthScope. Seismology has moved from a data-limited field to an analysis-limited field. The foundation of computational seismology is the generation of synthetic seismograms, used in the modeling of earth structure, earthquake rupture, and aspects of wave propagation. In the past, considerable simplifications were made in the calculation of seismic rays, usually made using ray theory that limited their applicability. Recently, an important advancement for global and regional seismology was made with the spectral element method (SEM) (Komatitsch *et al.*, 2002) where the propagation of seismic waves through the Earth can now be modeled accurately. The SEM takes into account heterogeneity in earth models, such as three-dimensional variations of seismic wave velocity, density, and crustal thickness. Although the full solution to the wave equation is computationally expensive, it has been implemented on parallel computers (Komatitsch *et al.*, 2002). This combination of hardware and software enables the simulation of broadband seismograms without intrinsic restrictions on the level of heterogeneity or the frequency content. These 3-D codes will revolutionize seismology, allowing for a direct investigation of countless

geodynamic topics (such as the fate of subducted lithosphere, existence of mantle plumes, lithospheric structure, plate boundary zone complexity) that are poorly suited for traditional “1-D” codes.

In the decades since the plate tectonics paradigm provided a framework for understanding the Earth, the earth sciences have changed markedly, as geoscientists have developed an increasing understanding of, and appreciation for, the range of scales in space and time that influence the planet’s evolution. Advances in technology, as well as more and better instrumentation, have led to breakthroughs in our understanding of the Earth’s structure and dynamics. Growing seismological databases have led to new, high-resolution models of the Earth’s interior, providing detailed images of structure from the crust to the inner core (Romanowicz, 2008). In the past decade, satellite geodesy has allowed us to observe active deformation within the crust on the human timescale. We are now being inundated by vast quantities of observational data on the Earth’s composition, structure, and dynamics. EarthScope is providing high-resolution seismic imaging of the North American continent through USArray, while the PBO component of EarthScope provides high-resolution, continuous geodetic observations of deformation along the western boundary of the North American plate.

5.2 Emerging from hero codes

Computer simulations came into wide use in geophysics during the decade after the plate tectonic revolution. Combining the new tectonic paradigm with high-quality observations allowed many problems to be addressed in novel ways by using computer models. Individual investigators responded with the development of self-contained research codes written from scratch for a narrow range of problems. Solution schemes and numerical algorithms that developed in other areas of science, most notably engineering, fluid mechanics, and physics, were adapted with considerable success to geophysics. Because the software was often largely the product of an individual effort, this style has informally been referred to as “heroic.” This approach has proven successful as is evident from the range of innovative papers using computer models starting in the 1970s; indeed, many of the examples presented above resulted from this programming model. For specific individual projects, this style of code development is certainly easiest, and will likely remain useful into the future. Nevertheless, its strength for solving problems of interest is now starting to show its limitations as we try to share codes and algorithms, or when we want to recombine codes in novel ways to produce new science.

As an example, consider *Citcom*, one of several sophisticated codes for solving mantle convection problems (Moresi and Gurnis, 1996; Moresi and Solomatov, 1995). Since its initial development phase in the early 1990s, *Citcom* is entirely

self-contained and written in C with its own unique data structures, mesh generators, input and output routines, and equation solvers for both heat transport and incompressible flow. Other convection, solid deformation or magma-migration codes have similar components but use different algorithms, physics, rheologies, or data structures, many of them tuned to meet the needs and interests of the individual investigator. Every one of these codes contains real gems of science and software. However, the ability to extract them, compare them, and recombine them into new science is currently daunting due to coding inconsistencies, language barriers, or poor documentation that were never designed to be publicly supported. It is clear to most modelers, that if we could change the way we do code development to a more modular, reusable approach, it would enable individual scientists to more readily use advances of other scientists while concentrating on the novel components of most interest to them.

This approach becomes crucial if we want to tackle multi-scale and multi-physics problems or combine additional techniques to calculate the observable consequences of dynamic problems. For example, few individual modelers have the time, background, or resources to write a comprehensive code that could calculate mantle convection and its seismic and geochemical signature. Moreover, there is a large community of users who would like to use simulation software for data interpretation, data assimilation, and hypothesis testing, but are not specialists in numerical methods. To make progress in these areas, train the next generation of modelers and geoscientists, and move the field forward as a whole requires that we change the way we develop scientific computations. This is not trivial and is unlikely to evolve spontaneously from the heroic model of code development. Given this backdrop, in 2005 the US National Science Foundation (NSF) sponsored such an effort – to move computational geodynamics forward – with the creation of a virtual organization, the Computational Infrastructure for Geodynamics (CIG).

5.3 The Computational Infrastructure for Geodynamics (CIG)

Computational Infrastructure for Geodynamics is an NSF center that develops computational software and maintains related services in support of earth sciences research. CIG is a community-governed organization with 46 Member Institutions (mostly research universities) and 10 Foreign Affiliates, under the control of an elected Executive Committee and administered by the California Institute of Technology. With a small team of software engineers, CIG develops, supports, and disseminates community-accessible software for the greater geodynamics community. The software is being developed for problems ranging widely from mantle and core dynamics, crustal and earthquake dynamics, magma migration, seismology, and related topics.

LAYER	Generic Geodynamic	Gale	PyLith	CitcomS
Superstructure	<ul style="list-style-type: none"> • Monitor Simulation • Couple Fluid to Solid • Visualization 	<ul style="list-style-type: none"> • Parallel VTK Output 	<ul style="list-style-type: none"> • Pyre Time Stepping 	<ul style="list-style-type: none"> • Pyre: Exchanger Time Stepping
Geodynamic Specific	<ul style="list-style-type: none"> • Rheology Modules • Assess plate tectonics 	<ul style="list-style-type: none"> • Rheologies • Free Surface 	<ul style="list-style-type: none"> • Rheology Modules • Assess plate tectonics 	<ul style="list-style-type: none"> • Rheology Built In
Infrastructure	<ul style="list-style-type: none"> • Meshes: Solid & Fluid • Solver: Solid & Fluid 	<ul style="list-style-type: none"> • GIGeMm: A framework for meshes, particles, solvers, boundary conditions, I/O ... 	<ul style="list-style-type: none"> • Meshing • Solve 	<ul style="list-style-type: none"> • Meshes: Cocom3 • Solver: Cocom3
Libraries	<ul style="list-style-type: none"> • Linear Algebra Solver 	<ul style="list-style-type: none"> • PETSc, MPI, libn2, UMFPACK, BLAS/LAPACK 	<ul style="list-style-type: none"> • PETSc, BLAS/LAPACK, MPI 	<ul style="list-style-type: none"> • PETSc, HDF5

Figure 5.1. Hierarchy of software used generally by CIG, as well as how this approach has been used in three community codes: *Gale*, *PyLith*, and *CitcomS*.

With a high level of community participation, CIG has attempted to leverage the state of the art in scientific computing into a suite of open-source tools and codes. The infrastructure consists of a number of different components. At its core is a coordinated effort to develop reusable, well-documented, and open-source geodynamics software. The development is made possible by a central system facilitating software development, including a repository, a bug tracking system, and automatic regression testing (described in the next section). The CIG software is organized in a hierarchy with the basic building blocks – an infrastructure layer – of software by which state-of-the-art modeling codes can be assembled (Figure 5.1). On top are extensions of existing software frameworks to interlink multiple codes and data through a superstructure layer. In addition to this hierarchy, we have also found it necessary to develop a Science Gateway to allow users to initiate and monitor simulations on the TeraGrid via the Web. CIG is more than software, and involves strategic partnerships with the larger world of computational science, as well as specialized training and workshops for both the geodynamics and larger earth-science communities.

CIG is a virtual organization governed by an Executive Committee (EC). The structure of CIG recognizes member institutions, which are educational and not-for-profit organizations with a sustained commitment to CIG objectives, and a number of foreign affiliate members. CIG has a Science Steering Committee (SSC) that consists of eight elected members that have been fully engaged in a dialog with the user community and active users of CIG software. The committee has a balance of expertise in both geoscience and computational sciences and provides guidance within all of the subdisciplines of computational geodynamics. Their principal duties are to assess the competing objectives and needs of all the subdisciplines covered by CIG, provide initial assessment of proposals submitted to CIG, and revise the Five-Year Strategic Plan. Recommendations from the SSC are passed on to the EC. Concepts and plans for CIG activities have come directly from the community, member institutions, working groups, and their elected committees.

Ideas and plans will move from members to the Science Steering Committee and finally to the Executive Committee. As part of the development of the Strategic Plan each year with a running five-year window, the SSC formulates a prioritized list of tasks for software development for the coming year, how these tasks are both interrelated and related to the broader needs of the community, and then transmits this as a recommendation to the EC.

CIG has established a small team of dedicated software architects and engineers whose work is guided by scientific objectives formulated by the scientific community. The Software Development Team (SDT) provides software services to the community in terms of programming, documentation, training, and support. Guidance for the programmers comes from the SSC.

5.4 How CIG develops software

In a virtual organization like CIG, the priorities for software development result from a dynamic balance between scientific needs, resources, and what is feasible technically. We hope to convey this balance below, where we describe specific software development activities. Here we focus on the infrastructure that has facilitated software engineering, including development, verification, and maintenance of the software.

An important aim is to introduce good software design practices into the software development efforts at CIG. This includes techniques for automated build and test procedures, development of benchmarks and test cases, and documentation. The software repository and attendant web site are central to CIG's objectives of facilitating collaboration and sharing of validated open-source software and reusable components. The repository is critical to bring modern software engineering practices to our community and CIG's software development team. Originally, CIG used a single repository for developer and community use that managed multiple developers working concurrently on modular software components shared through the repository. We use the open-source package Subversion (*SVN*) for the main CIG software repository, which contains most of CIG's codes. For the magma dynamics project (*MADDs*), CIG started to use the Mercurial repository (*hg*), although this is beyond the scope of this chapter. The entire contents of both repositories are navigable from our web site. In addition, CIG provides a bug-tracking database (*Roundup*) to allow developers and external participants to register and comment on bugs and requests for new functionality in CIG software that can then be worked on by the developers of a program. Finally, for each of the subdisciplines, we maintain a Listserve as well as editable web pages through our *Plone* site (like a Wiki site).

A key problem that faces any dynamic software repository is ensuring that "nothing breaks" despite frequent dynamic changes needed to meet the evolving

scientific goals of the community. CIG uses agile computing to minimize the risks of software development for continuously evolving requirements. In particular, the repository uses unit and regression testing. Building and testing in the *SVN* repository occurs either nightly or automatically in response to a software commit using *CIG-Regressor*, a collection of *Python* codes we wrote. This software uses *Buildbot*, extended by us, and the results of the testing are both stored in a database and made available interactively on our web site (Figure 5.2). Nightly regression testing generates an electronic report that contains the build and test failures (including the platforms on which they occurred). Regression testing allows the SDT to rapidly identify when a change in a repository component or platform has caused an error or inconsistency. Regression testing gives users of the repository confidence in the robustness of the software. We also extended *Buildbot* so that executable binaries for common platforms are automatically generated.

Implementing a comprehensive program of software verification has been an important aspect of CIG. This is a complex and rich topic, but had not been uniformly executed within computational geophysics. Software must first be verified, meaning that the software works as expected and that the equations solved give the expected results. However, the software must also give valid results, meaning that the physics and algorithms embodied in the software reproduce what occurs in either experiment or nature. CIG has mostly focused on software verification because much of software validation is related to the core of geodynamics research. Whether or not a certain computer code that has already been extensively verified adequately represents the physics of the underlying problem for which an algorithm has been designed has been a topic beyond CIG.

We used a multilevel verification plan. The solution from each *geophysical* solver (such as for Stokes flow) is compared against known analytical solutions to the governing equations (see Zhong *et al.*, 2008, for an example). For each solver that CIG produces, we make available at least one analytic solution that the solver can be automatically compared against. The solutions and prior results of these tests on different computers are published in our manuals and web page. In some cases, we have participated in community activities in which a range of numerical codes attempt to match the results of laboratory experiments.

To facilitate error analysis, benchmarking, and code verification, we developed *Cigma* (CIG Model Analyzer) that consists of a suite of tools for comparing numerical models. The current version of *Cigma* is intended for the calculation of L2 residuals in finite element models, but can be extended to compare arbitrary functions. In error analysis, *Cigma* calculates both the local error and a global measure of the differences between two solutions by performing integration over a discretized version of the domain. This comparison can take place even when the underlying discretizations do not overlap. In benchmarking, we hope that *Cigma*

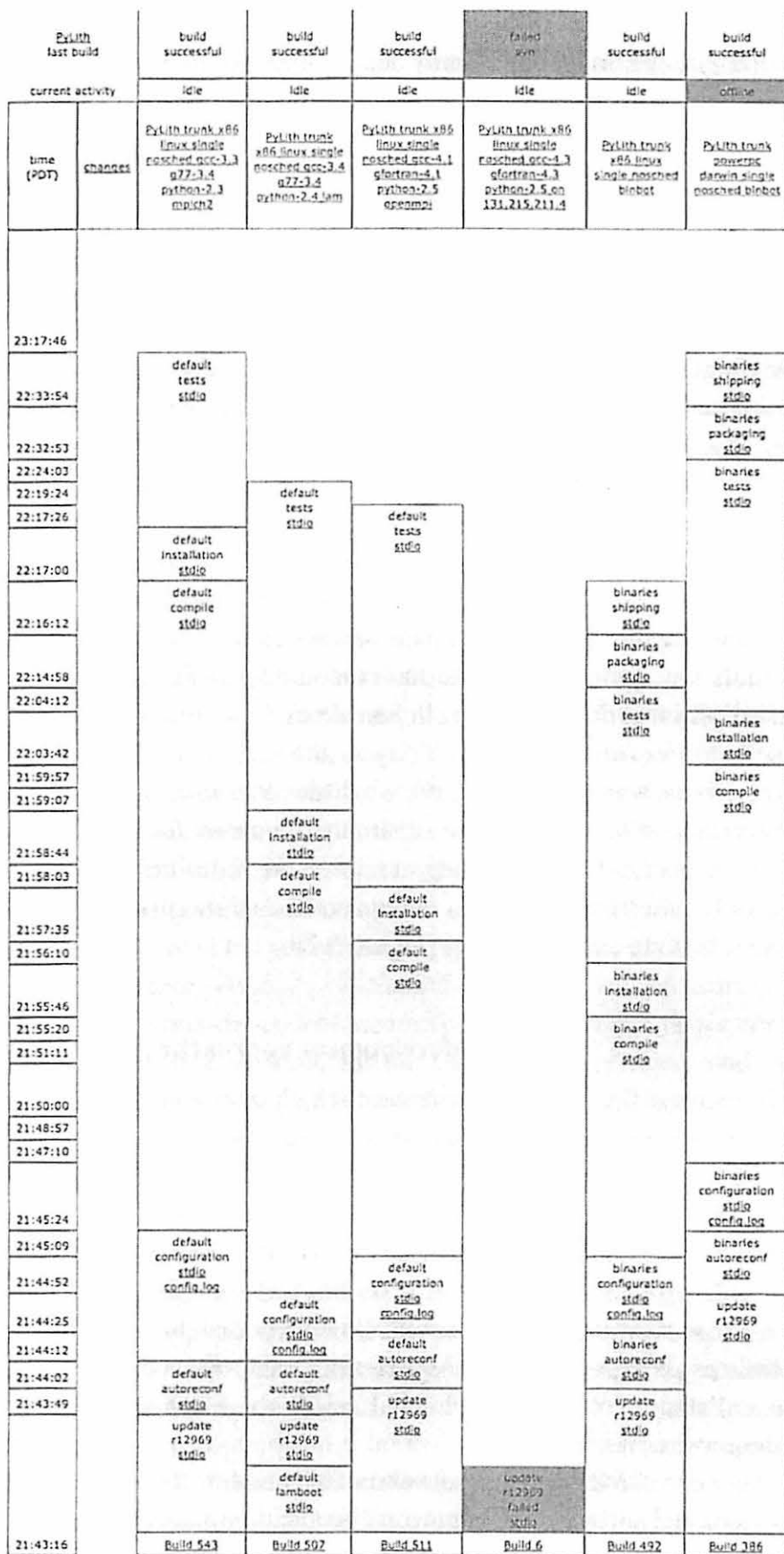


Figure 5.2. *Buildbot* automatically builds and tests code on a variety of platforms each time a change is made to the source. Here, *Buildbot*'s waterfall display indicates the status of some *PyLith* builds. Each column corresponds to a single platform.

will help the geodynamics community agree on a standard solution to specific problems by facilitating the process of comparing different numerical codes against each other. Lastly, as an automated tool, *Cigma* can help application developers create regression tests to ensure that software changes do not affect the consistency of the results. CIG developed *Cigma* in response to demand from the short-term tectonics community for a simple tool that can perform rigorous error analysis on their finite element codes. In the longer term *Cigma*, can be used for nearly all geodynamics modeling codes. *Cigma* relies on libraries. The Finite element Automatic Tabulator (FIAT) Python Library (FIAT, 2010) supports generation of arbitrary order instances of the Lagrange elements on lines, triangles, and tetrahedra. It can also generate higher-order instances of Jacobi-type quadrature rules on the same element shapes. The Approximate Nearest Neighbor (ANN) Library (ANN, 2010), written in C++, supports data structures and algorithms for both exact and nearest-neighbor searching in arbitrarily high dimensions. Both of these libraries extend and generalize *Cigma*'s functionality so it can handle other types of elements, and provide the ability to compare vector fields.

Of the tools we made available to the community to facilitate interaction, we found that activity was dominated by our Listserves. Few communities used our bug tracking software or our editable web pages, although there were notable exceptions. Several years into the life of CIG, we discovered that the ease-of-use of our software increased substantially if we distributed binaries for the packages, especially use that emerged from training sessions. We were able to streamline the production of binaries by automating their generation through our *Buildbot* system (a component that we called *Bin-bot*) (Figure 5.2).

5.5 Divergent development approaches

We quickly realized that even geodynamics, which one would have thought of as a rather small and homogeneous community, was in fact not so homogeneous. We found that each of the subdisciplines, as least within the United States, were distinct in terms of their technical expertise, their ability to develop codes, their reception of software engineering, and how close they were to achieving their goals with existing software. Some subcommunities were small but had enough experience in developing hero codes so that strong links with CIG never developed. While at another extreme, little experience existed to develop computational codes so that they were willing to collaborate with computational scientists. Most subcommunities fell between these extremes.

What we hope to convey are some details not only with software, but with community interactions and software development. The mantle convection community has a long history at developing and using codes. Two individual tectonics communities

illustrate the different outcomes possible when they are comfortable with a few individuals taking the lead in developing entirely new codes. Finally, we will describe our experience with computational seismology where we developed a Science Gateway using existing codes for broad community use.

5.5.1 An emerging community code in mantle convection

The mantle convection community has a long history of hero code development in which the products of software development have been passed on by generations of professors and graduate students. This community viewed the formation of CIG with suspicion since they nominally had the ability to develop codes and subsequently modify them for application to specific projects. CIG's primary role has been to bring standard well-proven codes under community control and then subsequently to add new features. CIG is now engaged in the development of a new package with adaptive mesh refinement (AMR) that transcends current capability (Burstedde *et al.*, 2008). We will review the functionality of the community code and give a brief history of its development.

Much of CIG's development for the mantle convection community has been with *CitcomS*, a finite element code written in C that solves for thermal convection within a spherical shell. The code is capable of solving many different kinds of convection problems using the flexibility of finite elements. The normal sequence of steps for the solution of convection problems starts with an initial temperature field. First, the momentum equation is solved. The solution of this equation gives us the velocity from which we then solve the advection–diffusion equation, giving a new temperature solution. Variable viscosity, including temperature-, pressure-, position-, composition-, and stress-dependent viscosity are all possible. This code uses an iterative solution scheme to solve for the velocity and pressure and, as such, a converged solution cannot be guaranteed. Nested inside the iterative scheme, the code uses either a conjugate gradient solver or a full multigrid solver to solve the discretized matrix equations.

The original development of *Citcom* in the early 1990s proceeded in much the same way as any hero code (Moresi and Solomatov, 1995). The original code turned out to be quite modular and easily extensible, and the fundamental finite element infrastructure is still in place and forms the basis for most versions of the software. A number of features were quickly added by a distributed user community: a three-dimensional Cartesian version (Moresi and Gurnis, 1996), a parallelized version using message passing routines on a limited release Intel supercomputer (Zhong *et al.*, 1998), a spherical version of the code named *CitcomS* (Zhong *et al.*, 2000) and a Beowulf implementation (Conrad and Gurnis, 2003), and many others. We quickly found ourselves with a plethora of versions of *Citcom* around the world

with different capabilities. Consequently, by 2002, there were so many different versions of the code that some rationalization was in order. The software was migrated into a version control and two versions, a fully spherical or regional model, were reengineered through the former GeoFramework project. By 2004, in order to increase the functionality of *CitcomS*, the developers began to reengineer the code into an object-oriented environment so that it could work with a *Python*-based modeling framework called *Pyre* (Cummings *et al.*, 2002) (Figure 5.1). The release of the software became known as *CitcomS.py*, and allowed multiple simulations to dynamically interact, such as a regional model within a global flow model (Tan *et al.*, 2006).

At this point, CIG took over the maintenance of the software for the community, but we found ourselves with a wide range of versions since the entire community neither used the same repository nor accepted the utility of software frameworks. Development of *CitcomS*, following the formation of CIG, proceeded in a somewhat similar fashion as occurring earlier with the GeoFramework project. Several members of the community had developed significant enhancements to the code that were not in the main repository trunk. Most notable was a particle tracking method that worked on top of the global mesh (McNamara and Zhong, 2004). Another was an alternative means by which to solve the equations for compressible flow. An example of combining the particle tracking with compressible convection is shown in Figure 5.3a. To increase functionality using common components, as illustrated in Figure 5.1, we incorporated the use of HDF5 (a parallel version of the Hierarchical Data Format). The most recent release of *CitcomS* (3.0) contains many new features, including two implementations of compressible convection; the ability to resume computation from previous checkpoints; multi-component chemical convection; a fixed non-Newtonian solver; and an exchanger package for solver coupling. By advocating the use of software repositories and allowing technically proficient users access to them, we have cut the number of alternative versions of the software down as well as shortening the time that features developed by others can be merged back into the main trunk.

CIG has also had an influence on the mantle convection community through workshops. We saw an important change in outlook between the two mantle convection workshops that we sponsored between 2005 (Boulder, CO) and 2008 (Davis, CA) (Table 5.1). What we observed during the first was a workshop dominated by the scientific questions, in which the computational methods were only reviewed. However, three years later, we saw an equally dynamic scientific discussion, but with discussion of new equation solvers and detailed attention paid to verification (benchmarking). We attribute this fundamental shift back to basics to the two mantle convection workshops and community organization between them, as shown in Table 5.1. A workshop on compressible mantle convection in Purdue, Indiana

Table 5.1. *Mantle convection and technical workshops*

Title	Location	Date	Attendance
2005 Mantle Convection Workshop	Boulder, CO	June 19–24, 2005	65
2006 Compressible Mantle Convection Workshop	Purdue, IN	March 27–28, 2006	14
Workshop on Challenges and Opportunities at the Interfaces of Scientific Computing	Austin, TX	October 16–17, 2007	64
AMR Tutorial Workshop	Boulder, CO	October 24–27, 2007	25
2008 Workshop for Advancing Numerical Modeling of Mantle Convection and Lithospheric Dynamics	Davis, CA	July 9–11, 2008	90
Workshop on Mathematical and Computational Issues in the Solid Earth Geosciences	Santa Fe, NM	September 15–17, 2008	55

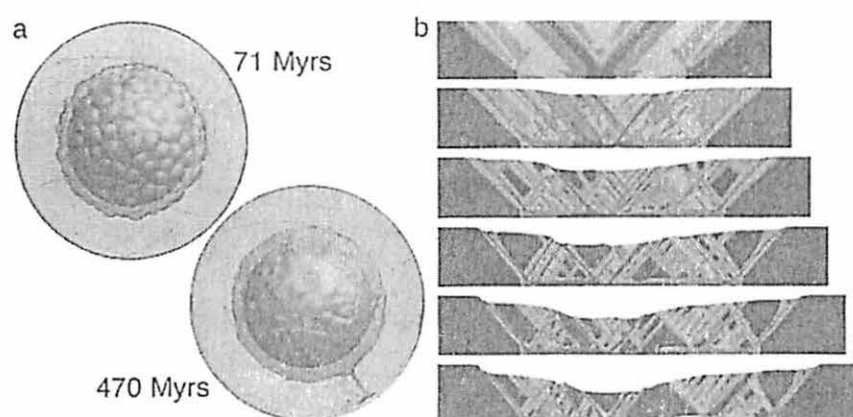


Figure 5.3. (a) Two time-instances in a calculation using *CitcomS* 3.0. The initial layer thickness is 250 km, and $Ra = 1.4 \times 10^9$. The density of normal mantle increases 50% from the surface to the CMB due to compressibility. The anomalous mantle has 8% larger bulk modulus and is 3.8% denser at the surface and 1% denser at the CMB. The viscosity is temperature and depth dependent, varying 400 \times due to temperature and is 10 \times larger for the lithosphere and the lower mantle. 120M tracers and $64 \times 64 \times 64 \times 12 = 3.1$ M elements and uses 48 processors. (b) Pictured is the strain-rate invariant for calculations of the GeoMod 2004 extension benchmark. The images range from 0 to 5 cm of extension, top to bottom. This model had a mesh of 512×32 and was computed on 32 processors using *Gale* 1.3.0. See color plates section.

focused on the development of a community code with new solvers, while a workshop on Challenges and Opportunities at the Interfaces of Scientific Computing in Austin, Texas brought investigators involved with new solver packages and meshing software into the broader discussion with geophysicists on how to advance the

science. We viewed this change in community outlook as essential, since detailed discussions of the computational challenges were essential for progress on the geophysical questions.

5.5.2 Starting from scratch in tectonics

Software development in tectonics resulted in two new codes. However, the software development proceeded in different ways and reflected different thrusts and technical abilities in two subcommunities. From the short-timescale tectonics (earthquakes) community, an existing working group with several individuals with computational experience teamed with CIG engineers to develop a new code from scratch. However, for long-timescale tectonics (orogenesis, basin formation), the community articulated its needs but was unable to take the lead developing the software to bring their vision to fruition. In that case, CIG teamed up with a group in Australia to produce a new code with common components.

While the planning for CIG was in its infancy, two members of the community, Brad Aagaard (USGS) and Charles Williams (RPI) began working towards integrating their individual codes, *EqSim* (Aagaard *et al.*, 2001) for dynamic rupture, and a version of *Tecton* (Melosh and Raefsky, 1980, 1981) for quasi-static problems), into the *Pyre* framework, with the ultimate goal of developing highly modular codes for the simulation of earthquake dynamics. A significant amount of commonality was identified between the codes, so that Aagaard and Williams then coordinated their development with a plan to merge their codes into a single suite of modules, *PyLith*. *PyLith* uses all of the levels in the software hierarchy that we first envisioned (Figure 5.1). As part of the general toolkit needed for the solution of many of the problems that CIG encounters, Matthew Knepley at Argonne National Laboratory (ANL) developed *Sieve*. *Sieve* is infrastructure for parallel storage and manipulation of general finite element meshes and can be used so that a developer avoids many of the complexities associated with parallel processing. *Sieve* is an integral component of the *PyLith*.

Developments in the long-timescale community proceeded quite differently. At a Tectonic Modeling workshop held in Breckenridge, Colorado, in June 2005, members of the tectonics community urged the development of a new open-source software code that could handle large deformations with viscoplastic rheologies. In particular, they advocated developing a code that uses the Arbitrary Lagrangian Eulerian (ALE) method. The ALE method uses a Eulerian grid to solve the Stokes flow problem and a Lagrangian grid to track material properties. Although such methods had long been used for 2-D problems, the expertise did not exist within the US community to develop a scalable 3-D code.

We realized that we could use technology developed by our partners, the Victorian Partnership for Advanced Computing (VPAC) in Australia, and develop such a code with common components. The end result was *Gale*, a parallel, two- or three-dimensional, implicit finite element code. The basic equations that *Gale* solves are the same as for mantle convection: Stokes and the energy equation. The development of *Gale* was jump-started by building on top of *Underworld* (Moresi *et al.*, 2003). *Gale* uses a hybrid particle-in-cell scheme, which combines a deformable mesh of computational points and a dense arrangement of mobile material points. The boundaries of the deformable mesh conform to the boundaries of the material as the simulation progresses, but the interior is constrained to remain as regular as possible. The particles track history-dependent properties such as strain for strain-softening materials. An example of a compression problem using *Gale* is shown in Figure 5.3b.

5.5.3 Production in a stable environment: An alternative for seismology

Development for the seismology community was very different than that for either mantle convection or tectonics. Arguably, for computational seismology the major underlying algorithms have already been developed and engineered into highly scalable codes (Komatitsch *et al.*, 2002). The frontier in this field, a field being deluged with data, is to bring the computational tools to the observational seismologist's workbench. Consequently, CIG developed a Science Gateway to allow observational seismologists to harness the power of computational codes running on remote supercomputers.

The CIG Seismology Web Portal enables the user to request synthetic seismograms for any given earthquake, selecting from an assortment of 3-D and 1-D earth models. Simulations are performed on the TeraGrid platforms. Upon completion of a simulation, the user receives a notification email directing them back to the web portal, where they can download the resulting seismograms in ASCII and SAC format (Figure 5.4). Visualizations include a graphic that depicts the earthquake's source mechanism, and maps showing the locations of the earthquake and the seismic stations.

The portal runs 3-D simulations using SPECFEM3D GLOBE, which simulates global and regional (continental-scale) seismic wave propagation using the spectral element method (Komatitsch *et al.*, 2002). A typical SPECFEM simulation runs from two to three hours, using 150 to 216 processors. The portal's 1-D simulations are performed by the serial *Mineos* code, which uses normal mode summation. To simulate an earthquake, the portal needs source information in Harvard CMT format. To obtain these input data, the user can search for events in

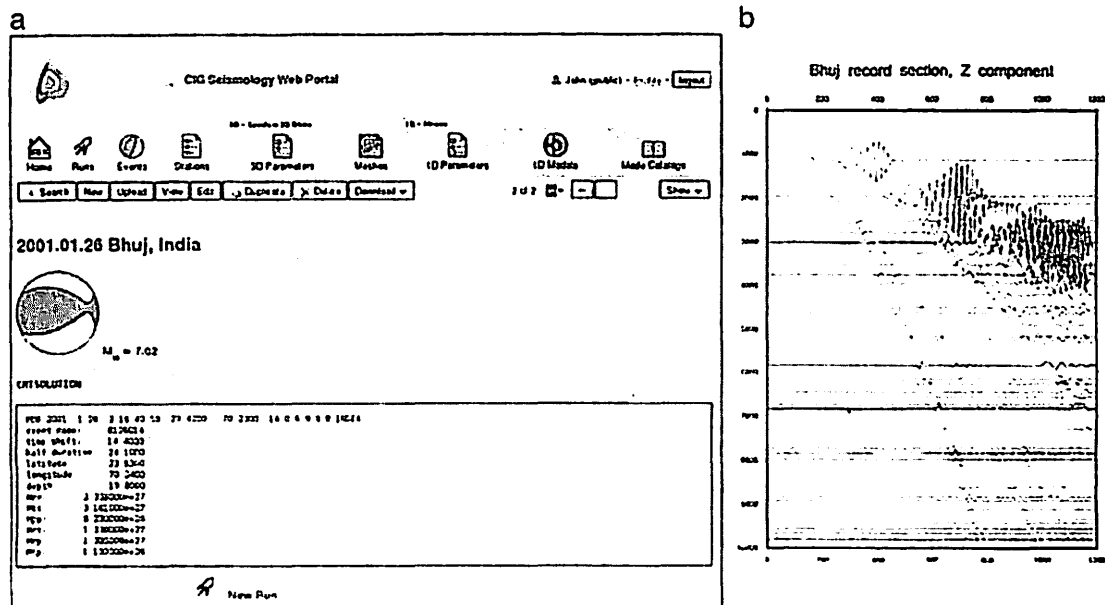


Figure 5.4. The CIG Seismology Science Gateway. (a) Screen shot. (b) Data created from the gateway, downloaded to a workstation, and displayed as a typical record section.

the database provided by the Global CMT Project. The database is integrated into CIG's web portal, allowing the user to select an earthquake by simply pointing and clicking. Alternatively, the user may upload custom CMT data to the portal. The portal provides a default set of seismic stations; the user may also upload a custom set.

The web site is written in *Python*, and built upon the *Django* web framework. Data persistence is achieved using an SQLite database. The site runs on top of Apache, which provides secure https connections. The web site is passive: it does not initiate connections to carry out tasks. Instead, the portal is powered by a separate daemon script that works in the background. The daemon constantly polls the web site looking for work to do. When a simulation request is posted, the daemon springs into action. First, it downloads input files for the simulation from the web site. Next, it connects to the TeraGrid cluster using GSI-Enabled OpenSSH and MyProxy credentials; then, it schedules and monitors an LSF batch job. Finally, it transfers simulation output files to a web server. Throughout this process, the daemon posts status updates to the web site using http POST, so that the user can monitor the progress of their request. The daemon script and other back-end helper scripts are also written in *Python*, but using the *Pyre* framework. The “beachball” and map visualizations are generated on the fly by the web server using GMT. The server to optimize performance caches the images.

5.6 Conclusions and future opportunities

As of this writing, CIG has finished its initial five years of community building and software development. An important component of the planning for the future was an assessment of the impact that CIG has had on geodynamics. In 2009, the geodynamics community produced an extensive document of science abstracts and statistics that painted a picture of major scientific advances using community developed software (Gurnis *et al.*, 2009). Research using CIG software has, for example, focused on enabling direct links between mantle convection and seismic observations, mineral physics, earth tides and gravity, and observations of surface vertical motions and sea level change, as well as elucidating the dynamics of plate tectonics, subduction, plumes, and the interior of other planets. CIG software has been used to improve images of the Earth's interior through adjoint tomography, directly test tomographic models with seismic waveforms, and improve and automate moment tensor inversions of earthquakes. It has also been used in some of the first three-dimensional models of the initiation and growth of faults in extensional tectonic environments. CIG software has facilitated studies of the interplay of crustal extension and melting, hill slope failure, and surface inflation associated with volcanic intrusion.

As the CIG community effort goes forward, it is essential that we modify how we do business, develop software, and interact as a community. Perhaps the most important change now taking place is to develop software using a common set of libraries. Adaptive mesh refinement has emerged as an important area where more emphasis is needed and where important progress has been made over the last several years. One possibility is to exploit the methods used for large mantle convection problems using octree meshes, which can be scaled to 10^5 processors and allow resolutions as small as 1 km here needed (Burstedde *et al.*, 2008), to be applied broadly for convection, crustal dynamics, and magma dynamics problems. As the CIG community effort moves its central site from Caltech to the University of California, Davis, we are optimistic that the rapid pace of advancement in the use and application of community-developed and applied geodynamics software will continue.

References

- Aagaard, B., Heaton, T., and Hall, J. (2001). Dynamic earthquake ruptures in the presence of lithostatic normal stresses, implications for friction models and heat production. *Bulletin of the Seismological Society of America*, **91**: 1765–1796.
- ANN (2010). www.cs.umd.edu/~mount/ANN/.
- Bercovici, B., Ricard, Y., and Richards, M.A. (2000). The relation between mantle dynamics and plate tectonics: A primer. In *History and Dynamics of Global Plate*

- Motions*, ed. M. A. Richards, R. G. Gordon and R. D. van der Hilst. Geophysical Monograph 121. American Geophysical Union, pp. 5–46.
- Burstedde, C., Ghattas, O., Gurnis, M. *et al.* (2008). Scalable adaptive mantle convection simulation on petascale supercomputers. International Conference for High Performance Computing, Networking, Storage, and Analysis, ACM/IEEE Supercomputing.
- Conrad, C. P. and Gurnis, M. (2003). Seismic tomography surface uplift, and the breakup of Gondwanaland: Integrating mantle convection backwards in time. *Geochemistry Geophysics Geosystems*, 4: 1031, doi:10.1029/2001GC000299.
- Cummings, J., Aivazis, M., Samtaney, R. *et al.* (2002). A virtual test facility for the simulation of dynamic response in materials. *Journal of Supercomputing*, 23: 39–50.
- Davies, G. F. (1999). *Dynamic Earth: Plates, Plumes and Mantle Convection*. Cambridge: Cambridge University Press.
- FIAT (2010). www.fenics.org/wiki/FIAT.
- Gurnis, M. *et al.* (2009). *Expanding Computational Infrastructure: The First 5 Years of CIG*. Pasadena, CA: Computational Infrastructure for Geodynamics, <http://geodynamics.org/cig/proposals/docs/documents/cig-first-5-years>.
- Karner, G. D., Taylor, B., Driscoll, N. W., and Kohlstedt, D. L. (2004). *Rheology and Deformation of the Lithosphere at Continental Margins*. New York: Columbia University Press.
- Komatitsch, D., Ritsema, J., and Tromp, J. (2002). The spectral-element method: Beowulf computing and global seismology. *Science*, 298: 1737–1742.
- McNamara, A. K. and Zhong, S. (2004). Thermochemical structures within a spherical mantle: Superplumes or piles? *Journal of Geophysical Research*, 109: doi:10.1029/2003JB002847.
- Melosh, H. J. and Raefsky, A. (1980). The dynamical origin of subduction zone topography. *Geophysical Journal of the Royal Astronomical Society*, 60: 333–354.
- (1981). A simple and efficient method for introducing faults into finite element computations. *Bulletin of the Seismological Society of America*, 71: 1391–1400.
- Moresi, L. N. and Gurnis, M. (1996). Constraints on the lateral strength of slabs from three-dimensional dynamic flow models. *Earth and Planetary Science Letters*, 138: 15–28.
- Moresi, L. N. and Solomatov, V. S. (1995). Numerical investigation of 2D convection with extremely large viscosity variations. *Physics of Fluids*, 7: 2154–2162.
- Moresi, L. N., Dufour, F., and Muhlhaus, H.-B. (2003). A Lagrangian integration point finite element method for large deformation modeling of viscoelastic geomaterials. *Journal of Computational Physics*, 184: 476–497.
- Moresi, L., Gurnis, M., and Zhong, S. (2000). Plate tectonics and convection in the Earth's mantle: Toward a numerical simulation. *Computing in Science and Engineering*, 2: 22–33.
- Ranalli, G. (1995). *Rheology of the Earth*. New York: Chapman and Hall.
- Romanowicz, B. (2008). Using seismic waves to image Earth's internal structure. *Nature*, 447: 198–201.
- Schubert, G., Turcotte, D. L., and Olson, P. (2001). *Mantle Convection in the Earth and Planets*. Cambridge: Cambridge University Press.
- Tan, E., Choi, C., Thoutireddy, P., Gurnis, M., and Aivazis, M. (2006). GeoFramework: Coupling multiple models of mantle convection within a computational framework. *Geochemistry Geophysics Geosystems*, 7, Q06001, doi:10.1029/2005GC001155.
- Zhong, S., Gurnis, M., and Moresi, L. (1998). The role of faults, nonlinear rheology, and viscosity structure in generating plates from instantaneous mantle flow models. *Journal of Geophysical Research*, 103: 15 255–15 268.

- Zhong, S., McNamara, A., Tan, E., Moresi, L., and Gurnis, M. (2008). A benchmark study on mantle convection in a 3-D spherical shell using CitcomS. *Geochemistry Geophysics Geosystems*, 9, Q10017, doi:10.1029/2008GC002048.
- Zhong, S., Zuber, M. T., Moresi, L. N., and Gurnis, M. (2000). The role of temperature-dependent viscosity and surface plates in spherical shell models of mantle convection. *Journal of Geophysical Research*, 105: 11 063–11 082.

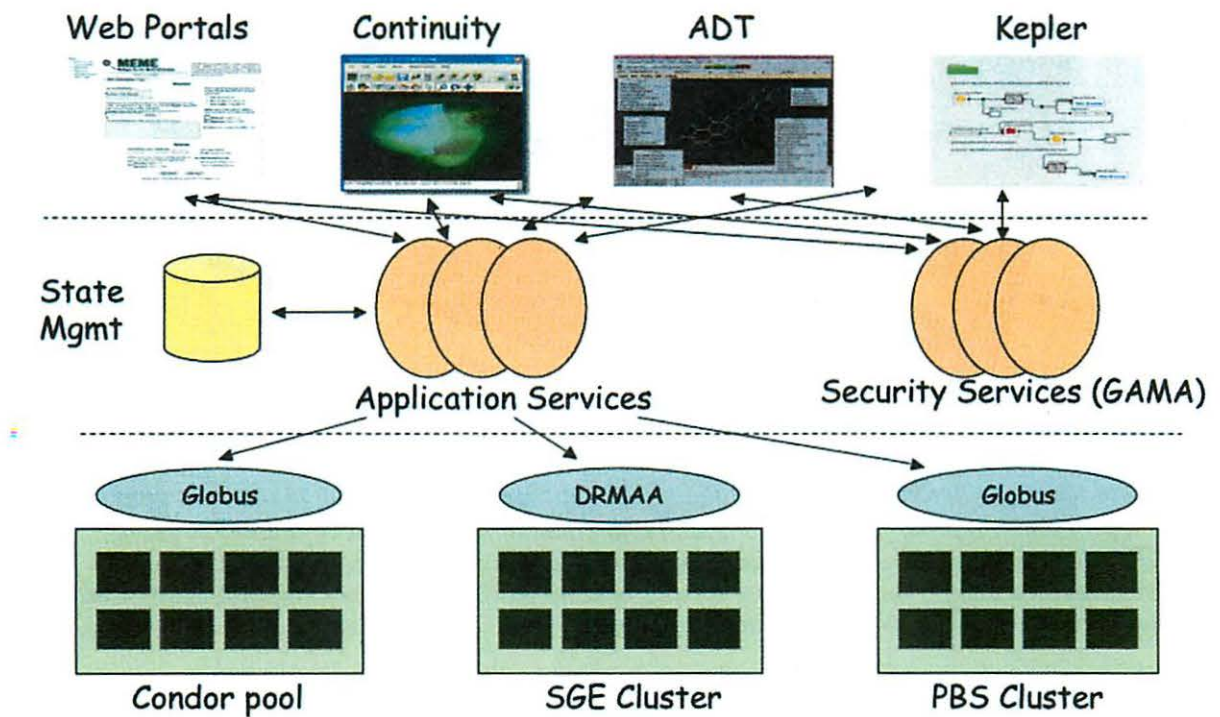


Plate 3. NBCR SOA architecture. (See Figure 4.1.)

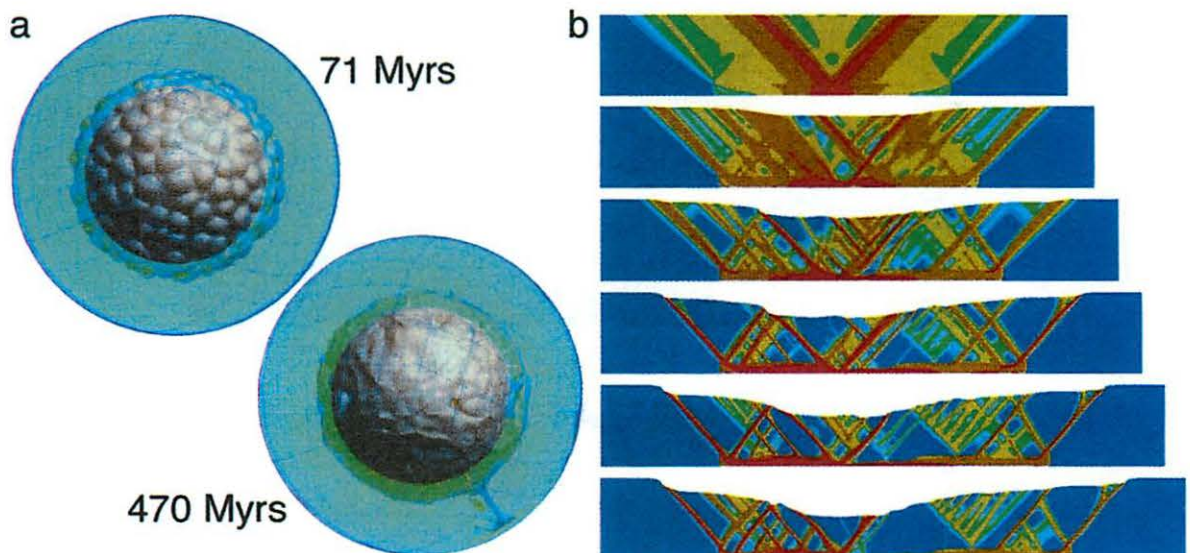


Plate 4. (a) Two time-instances in a calculation using *CitcomS* 3.0. The initial layer thickness is 250 km, and $Ra = 1.4 \times 10^9$. The density of normal mantle increases 50% from the surface to the CMB due to compressibility. The anomalous mantle has 8% larger bulk modulus and is 3.8% denser at the surface and 1% denser at the CMB. The viscosity is temperature and depth dependent, varying 400 \times due to temperature and is 10 \times larger for the lithosphere and the lower mantle. 120M tracers and $64 \times 64 \times 64 \times 12 = 3.1M$ elements and uses 48 processors. (b) Pictured is the strain-rate invariant for calculations of the GeoMod 2004 extension benchmark. The images range from 0 to 5 cm of extension, top to bottom. This model had a mesh of 512×32 and was computed on 32 processors using *Gale* 1.3.0. (See Figure 5.3.)